IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

Design and Implementation of a Digital Function Signals Generator using FPGA

Dr. Kamal Aboutabikh[1], Dr. Abdul Aziz Shokyfeh[2], Dr. Amer Garib[3]

Faculty of Informatics Engineering, Ittihad Private University, Damascus Syria[1,2,3.]

Abstract: In this paper, the direct digital frequency synthesizer (DDS) specifications were improved by increasing the number of accumulator bits, increasing the memory capacity of the generated signal samples, using a DAC by increasing the number of its bits, as well as generating different types of analogue signals in a digital way.

In this paper also, we discuss a practical mechanism of a digital function signals generator (DFSG) based on a direct digital frequency synthesizer (DDFS) using Cyclone II EP2C20F484C7 FPGA from ALTERA placed on education and development board DE-1 with the following parameters:

-Output waveforms: Sin, Gaussian, Sinc, Square , Saw tooth ,Triangular, and Wight Noise Signals.
-Frequency range: (3Hz…..10000 KHz).
-Frequency Resolution (3Hz).
-Signal amplitude (5V).
-With Reset the generator.
-Frequency of the generated signal for all types (1MHz).

Keywords: DFGS, DDFS , FPGA, DPNG , SIN, SINC ,SQUARE, SAWTOOTH, GAUSSIAN, TRIANGULAR .

I. INTRODUCTION

Forming signals with different types (sine, triangle, sawtooth, Square,…) using analogue methods (as it known) depends on using familiar elements as transistors, ICS, resistances, coils, capacitors, control elements, and adjusting and calibrate elements.

To change the parameters of this signals (frequency, amplitude, phase), we must change the value of the elements that form this signals, while the digital method which form these signals depends on calculating the samples value of the formed signal during one period by using the mathematical equation which describe the desired signal, and store this values into digital memory to read it serially from this memory with every fixed frequency clock pulse, which determine the frequency range of the frequency synthesizer with well-known Shannon Equation.

Frequency synthesizers have feature that we can change the parameters of the signal by changing the accessing method to the ROM that contain the formed signal samples values (amplitude, phase) and also apply mathematical processes on this values to change the amplitude and make the necessary modification as we will see in this paper.

In this paper a Digital Function Generator have been designed as an application on DDFS.

In Paper [1], the DDS is designed for only simulation with the following specifications:

-sinusoidal signal.
-Frequency Range: (0Hz…..160 KHz).
-Frequency Resolution (5Hz).

In Paper [2], the author focused on reducing the size of the memory allocated to store the values of the generated signal samples, but in this paper we do not care about the size of the memory because modern FPGA chips have a huge memory size. Rather, we considered that increasing the size of the memory increases the accuracy of shaping the digital signal and improves its specifications significantly.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

II. RESEARCH IMPORTANCE AND ITS OBJECTIVES

The importance of the research stems from the urgent need to use a digital frequency synthesizer in some applications, measurement systems, and modern communication systems, as some applications require great accuracy in the value of the generated frequency and high frequency switching rates, especially in frequency hopping systems used in wireless communications, and this is what can be achieved through DDFS, in addition to that, DDFS gives great flexibility in design and implementation, also is useful in greatly shortening the circuits and electronic parts used in implementation, which facilitates the maintenance process and reduces size, weight and cost. It also reduces power consumption and the emission of heat that is difficult to dissipate in some applications.

III. RESEARCH MATERIALS AND ITS WAYS

To design, and test the DFSG for different types of signals, the following tools and software are used:

- Cyclone II EP2C20F484C7 FPGA chip from ALTERA with highly accuracy, speed, and level specifications, placed on education and development board DE-1 [3].

-DDFS which is considered as highly accuracy techniques in Sinusoidal ,
Sawtooth ,Triangular ,Square ,Gaussian ,Sinc and noise signals synthesizing on FPGA chips.

-VHDL programming language with Quartus II.1 design environment [4].

- Design Environment MATLAB R2008a

-GDS-1052 digital oscilloscope with Free Wave program to take the results.

-PC computer for designing and injecting the design in the chip FPGA .

IV. BLOCK DIAGRAM OF THE DIGITAL FUNCTION SIGNALS GENERATOR (DFSG) AND SPECIFICATIONS

The functional diagram of the DFSG is shown in figure (1) and is composed of:

1-Direct Digital Frequency Synthesizer (DDFS) with the following specifications:

- An accumulator with an input of (n = 24) bits and an output of (b= 13 bits), which is allocated to form the reading address from the ROM memory, according to the value of the frequency to be formed, the value of (n), and the value of the clock pulse frequency used, according to the basic mathematical relationship of the digital synthesizer[5]:

$$f_{OUT} = \frac{F_{CLK} * L}{2^n} \Rightarrow L = \frac{2^n f_{OUT}}{F_{CLK}} \qquad (1)$$

Or :

$$T_{OUT} = \frac{1}{f_{OUT}} = \frac{2^n}{F_{CLK} * L} \Rightarrow L = \frac{2^n}{T_{OUT} * F_{CLK}}$$

Where:

(f_{OUT}) : output frequency of DDFS , (T_{OUT}) output period of DDFS.

(n): the number of phase accumulator (bits).

(L): frequency code of output frequency (f_{OUT}).

DOI: 10.17148/IJARCCE.2024.13503

F_{CLK}: frequency of clock pulses generator.

-The first ROM_SIN (with a capacity of 8KB) contains (8192X8 bits) samples of the sinusoidal signal during one cycle.

-The second ROM_ TRIANGULAR contains (8192X8 bits) samples of the triangular signal during one cycle.

-The third ROM_ GAUSSIAN contains (8192X8 bits) samples of the triangle signal during one cycle.

-The fourth ROM_ SINC contains 8192X8bit samples of the pulse signal during one cycle.

2-Digital pass-through (bass) voter with four channels (7 (8bits} x 1) to choose the signal type.

3-Digital-analogue converter (DAC) with 8 bits input to convert memory sample values from digital to analogue form.
4-Sawtooth Signal generator.

5-Square pulse generator.

6-Noise signal generator.

7-50MHz clock pulse generator.

Specifications of DFSG

-Signal types is: Sin, Sinc, Gaussian, Triangular, Square, Sawtooth and Noise signals.

- $F_{CLK} = 50MHz$.

-Frequency Range: (3Hz…..5000 KHz)

-Frequency Resolution $\delta(\mathrm{f})$ [5]:

$$\delta(\mathrm{f}) = \frac{F_{CLK} * L}{2^n} \qquad (2)$$

For L=1 we get :

$$\delta(\mathrm{f}) = \frac{F_{CLK} * L}{2^n} = \frac{50*10^6*1}{2^{24}} = 3HZ$$

-Frequency of the generated signal for all signals types equal ;(1MHz):

For f_{out}=1MHz we get :

$$L = \frac{2^n f_{OUT}}{F_{CLK}} = \frac{2^{24}*1*10^6}{50*10^6} = 335544$$

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

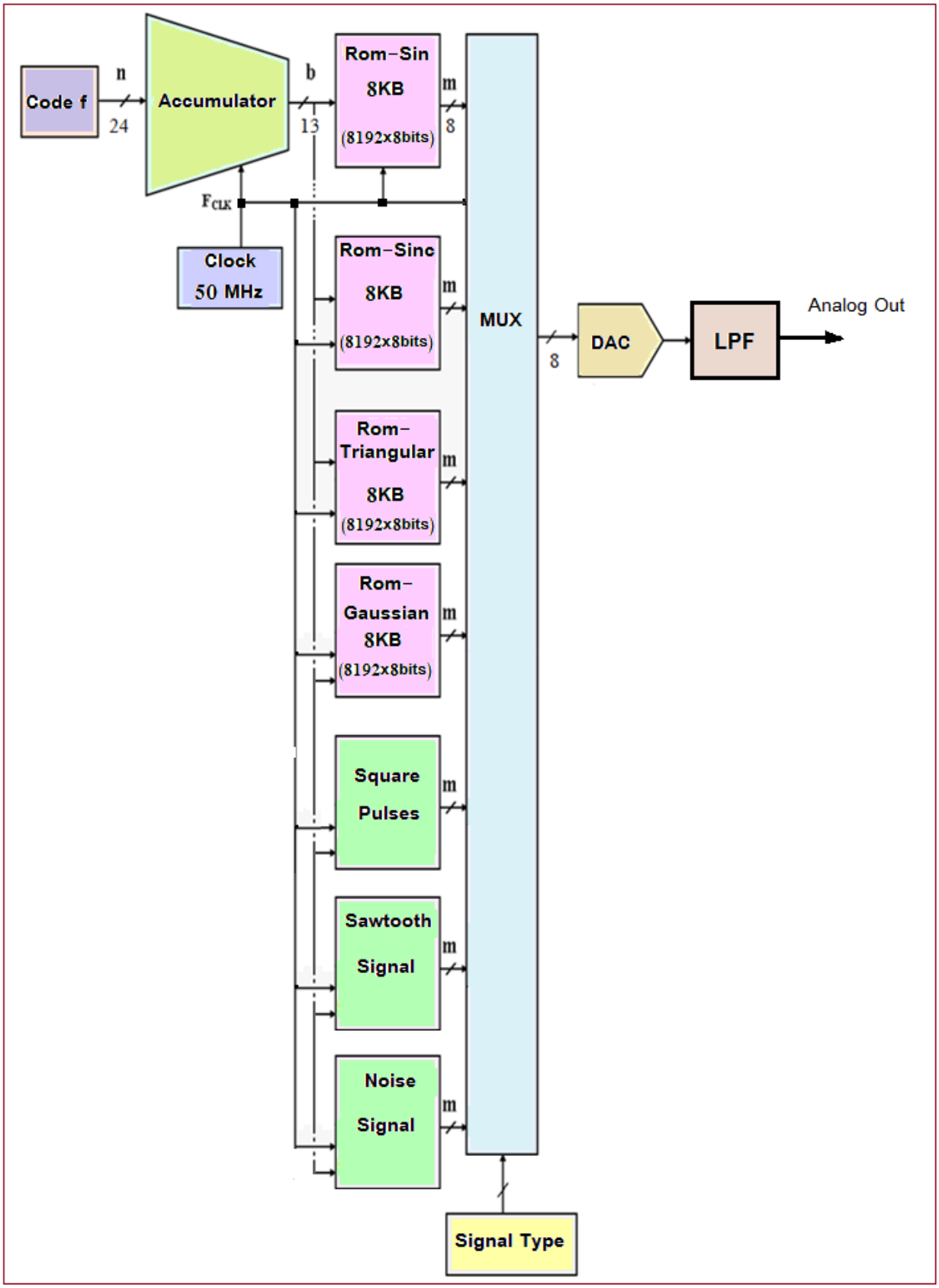

Fig. 1: Functional diagram of the DFSG.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

V. THE BLOCK DIAGRAM OF THE SINUSOIDAL SIGNAL GENERATOR AND SPECIFICATIONS

The stored values of the sine signal in ROM are calculated according to the relationship [5]:

$$U_{SIN}(i) = INT\left[\left(2^{m-1}-1\right).Sin\left(\frac{360.i}{2^{b}}\right)\right] \qquad (3)$$

For:

$$\mathrm{m} = 8\,\mathrm{bits}\,,\ \mathrm{b} = 13\,\mathrm{bits}\,,\ i = (0\ldots 2^{b}-1) = (0\ldots 8191)$$

Then

$$U_{SIN}(i) = INT\left[\left(2^{8-1}-1\right).Sin\left(\frac{360.i}{2^{13}}\right)\right] = INT\left[127.Sin\left(\frac{360.i}{8192}\right)\right]$$

$$U_{SIN}(i) = (-127.....+127)$$

To avoid negative values that are difficult to store in memory, we add a value off-set= 128

$$U_{SIN}(i) = (0.....255)$$

To obtain samples of the sinusoidal signal, we use (MATLAB R2008a) according to the following program section for one period:

```
i=0:1:8191;
y=128+floor(127*sin(2*pi*i/8192));
plot(i,y);
grid;
title('A sin signal');
xlabel('Time');
ylabel('Amplitude')
```

Then we store these values in a notepad file and load it into the sin signal generator's ROM.

The design result of the sinusoidal signal using MATLAB program is shown on the figure (2).

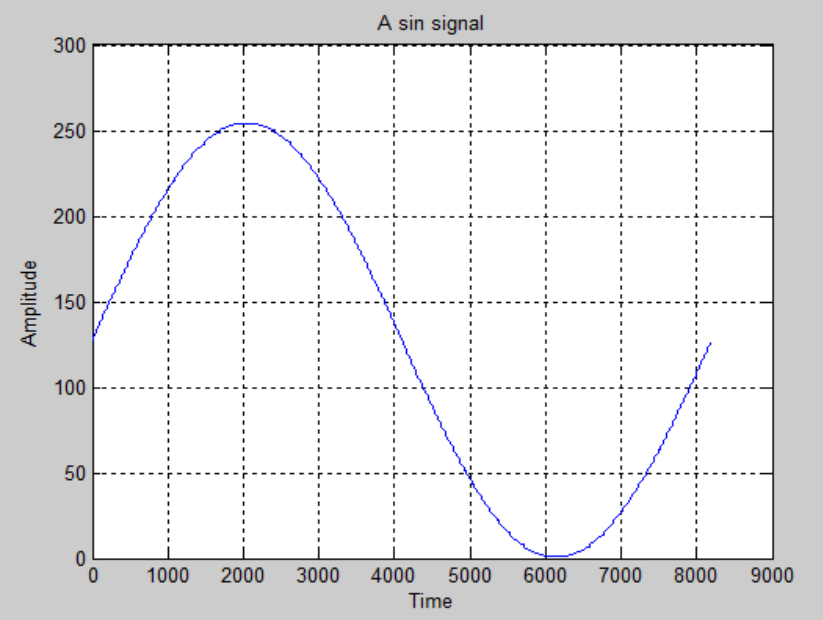

Fig. 2: Sinusoidal signal using MATLAB program for one period.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

The block diagram of the Sinusoidal signal Generator in Quartus II 9.1 design environment is shown in figure (3) [6].

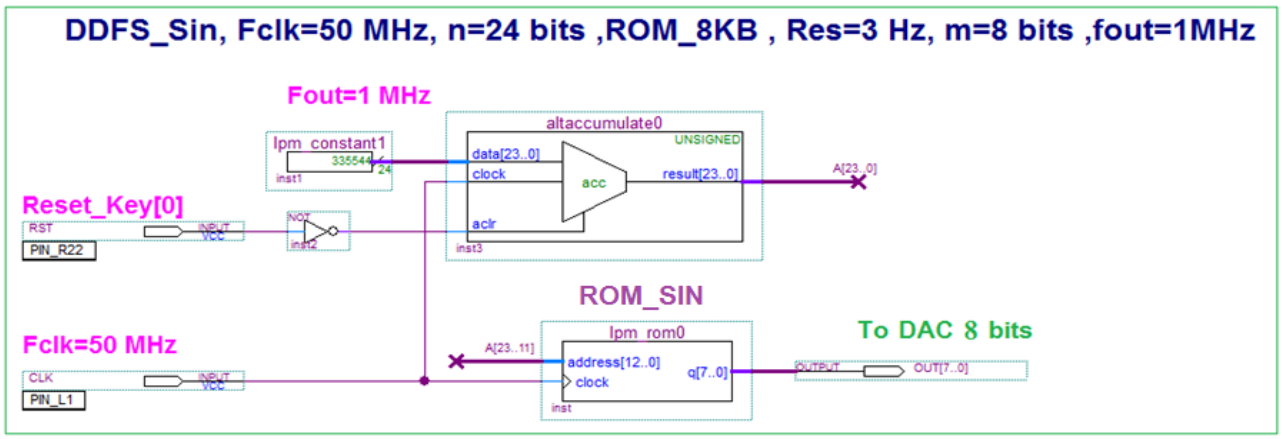

Fig. 3: Block diagram of the Sin Signal generator.

The design result of the sinusoidal signal generator is taken from the digital oscilloscope screen and shown on the figure (4) for f_{out}=1MHz.

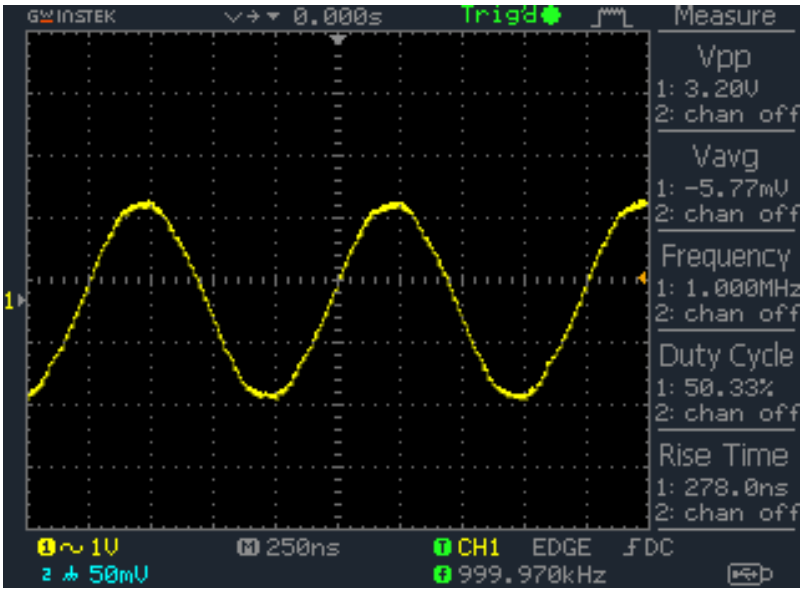

Fig. 4: The sin signal on the digital oscilloscope screen.

VI. THE BLOCK DIAGRAM OF THE SINC PULSE GENERATOR AND SPECIFICATIONS

The values of the Sinc pulse that are stored in ROM are calculated according to the relationship [5]:

$$U_{SINC}(i) = \frac{2^m}{2} + INT\{(\frac{2^m}{2} - 1) * \frac{\sin(\pi * i / T)}{(\pi * i / T)}\}$$

$$U_{SINC}(i) = \frac{2^m}{2} + INT\{(\frac{2^m}{2} - 1) * \sin c(\pi * i / T)\}$$

$$i = \{-(2^b - 1).....2^b\} \qquad (4)$$

$$U_{SINC}(i) = (0.....2^m - 1)$$

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

For:

$$m = 8\ bits\ , b = 13\ bits, T = 1$$

Then:

$$i = (-4095.....4096)$$

$$U_{\sin c}(i) = 128 + INT\left\{127 * \frac{\sin(\pi * i / 4096)}{(\pi * i / 4096)}\right\}$$

$$U_{\sin c}(i) = 128 + INT\{127 * \sin c(\pi * i / 4096)\}$$

$$U_{\sin c}(i) = (0.....+ 255)$$

To obtain samples of the sinc pulse, we use MATLAB according to the following program section for (T=1):

```
i= - 4095:1:4096;
y=(pi*i/4096);
z=128+floor(127*sinc(y));
plot(i,z);
grid;
title('A sinc signal');
xlabel('Time');
ylabel('Amplitude')
```

Then we store these values in a notepad file and load it into the sinc pulse generator's ROM.
The design result of the sinc pulse using (MATLAB R2008a) program is shown on the figure (5-a) for (T=1) and on the figure (5-b) for (T=2/3).

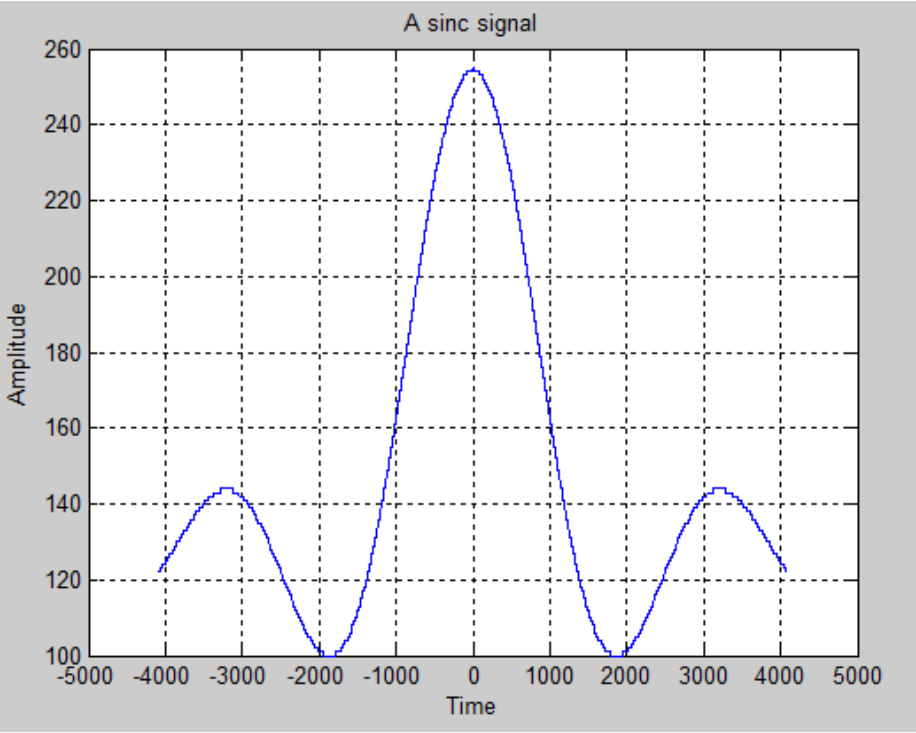

Fig. 5-a: Sinc pulse using MATLAB program for (T=1).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

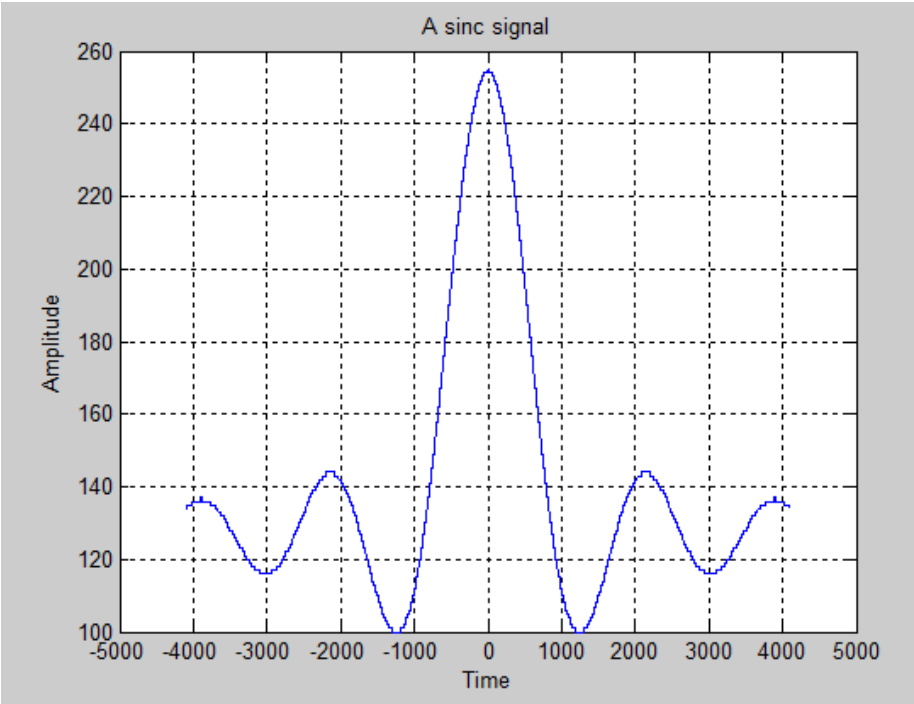

Fig. b-a: Sinc pulse using MATLAB program for (T=2/3).

The block diagram of the sinc pulse Generator in Quartus II 9.1 design environment is shown in figure (6).

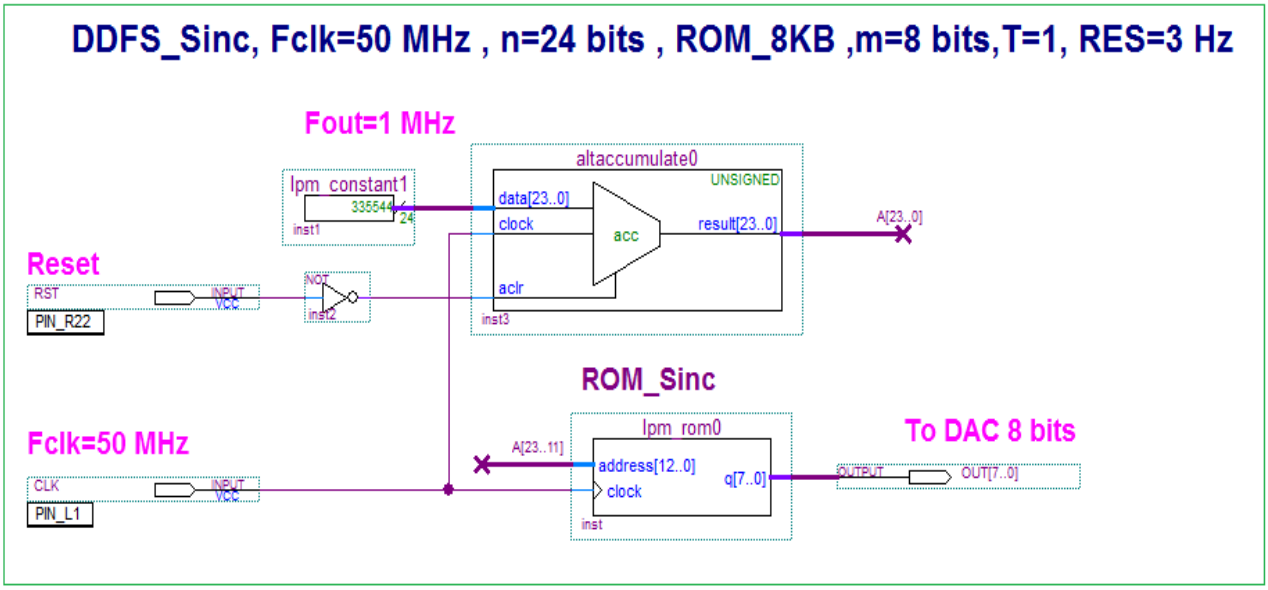

Fig. 6: Block diagram of the Sinc pulse generator.

The design result of the sinc pulse generator is taken from the digital oscilloscope screen and shown on the figure (7-a) for (T=1, f_{out}=1MHz) and on the figure (7-b) for (T=2/3, f_{out}=1MHz)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

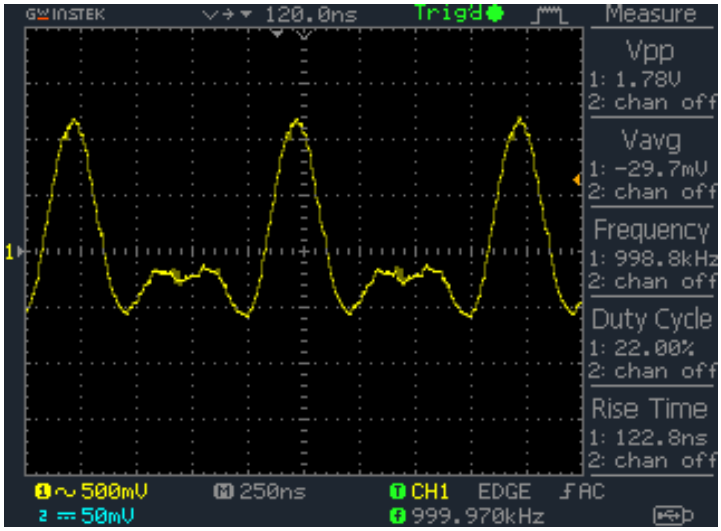

Fig. 7-a: The sinc pulse on the digital oscilloscope screen for (T=1, f_{out}=1MHz).

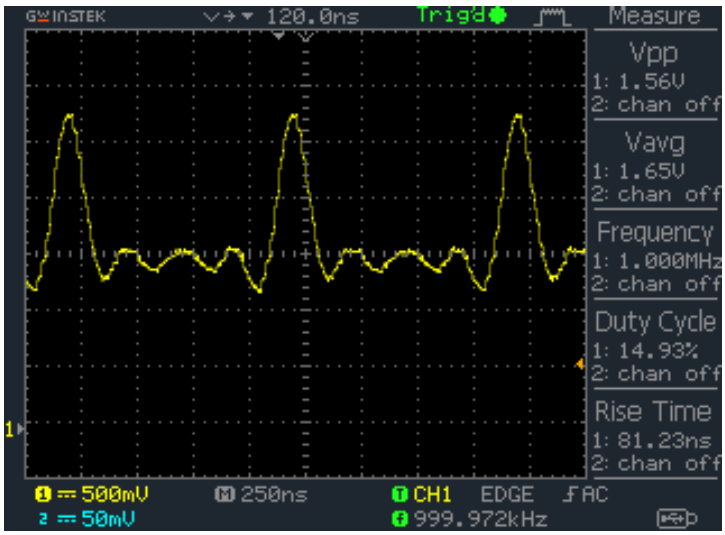

Fig. 7-b: The sinc pulse on the digital oscilloscope screen for (T=2/3, f_{out}=1MHz).

VII. THE BLOCK DIAGRAM OF THE GAUSSIAN PULSE GENERATOR AND SPECIFICATIONS

The values of the Gaussian pulse that are stored in ROM are calculated according to the relationship [5]:

$$U_{GAUS}(i) = INT\{(2^m - 1) * \exp(-a^2 * i^2 / x^2)\}$$

$$i = \{-(\frac{2^b}{2} - 1).....0,.....,\frac{2^b}{2}\} \qquad (5)$$

$$x = \frac{a * (2^b / 2)}{3}$$

$$U_{GAUS}(i) = (0.....2^m - 1)$$

For:

$$\alpha = 1, \mathrm{m} = 8\,\mathrm{bits}, \mathrm{b} = 13\,\mathrm{bits}, \; i = (-4095.....+4096)$$

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

Then:

$$x = \frac{a*(2^b/2)}{3} = \frac{1*(2^{13}/2)}{3} = 4096/3 = 1365$$

$$U_{GAUS}(i) = INT\{255*\exp(-i^2/(1365)^2)\}$$

$$U_{GAUS}(i) = (0.....255)$$

To obtain samples of the Gaussian pulse, we use MATLAB according to the following program section for (a =1):

```
i= - 4095:1:4096;
x=power(i,2);
y=power(1365,2);
z=exp(-x/y);
g=floor(255*z);
plot(i,g);
grid;
title('A Gaussian pulse');
xlabel('Time');
ylabel('Amplitude')
```

Then we store these values in a notepad file and load it into the Gaussian pulse generator's ROM.

The design result of the Gaussian pulse using (MATLAB R2008a) program is shown on the figure (8-a) for (a =1) and on the figure (8-b) for (a =2).

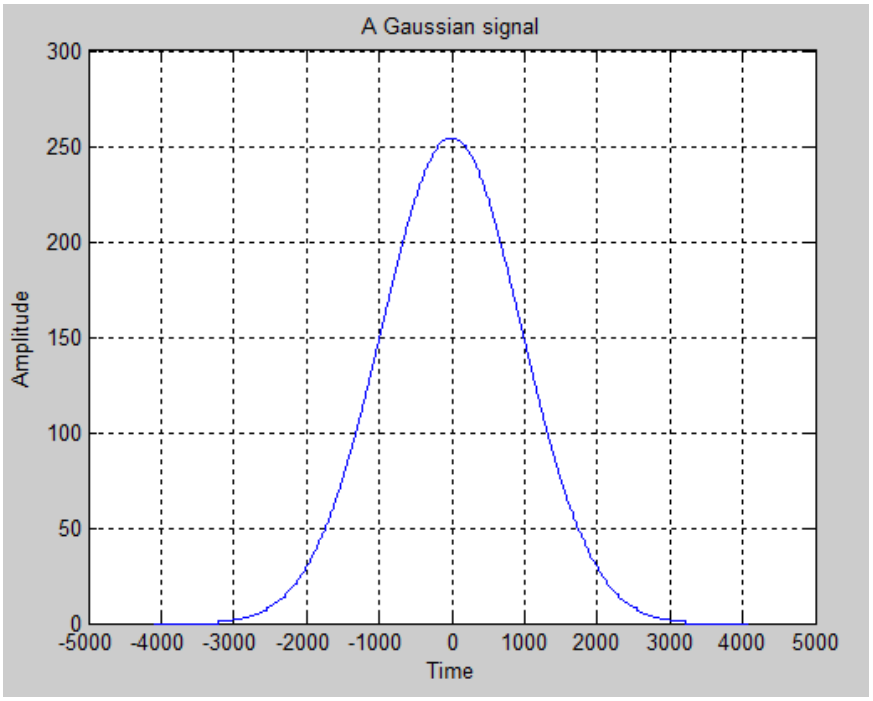

Fig. 8-a: Gaussian pulse using MATLAB program for (a =1).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

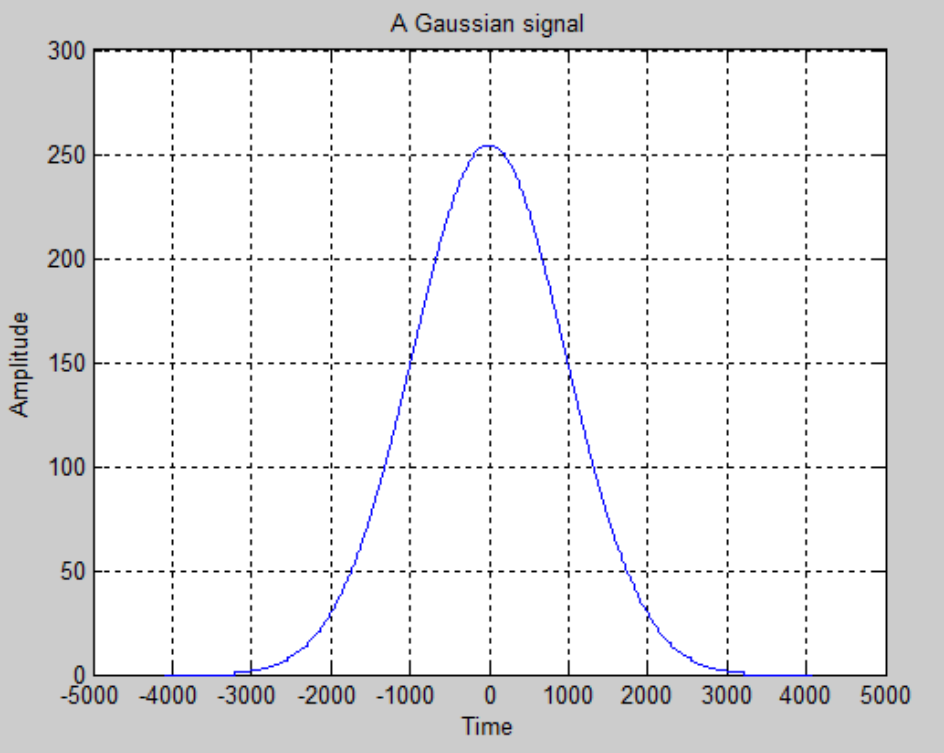

Fig. 8-b: Gaussian pulse using MATLAB program for (a =2).

The block diagram of the Gaussian pulse Generator in Quartus II 9.1 design environment is shown in figure (9).

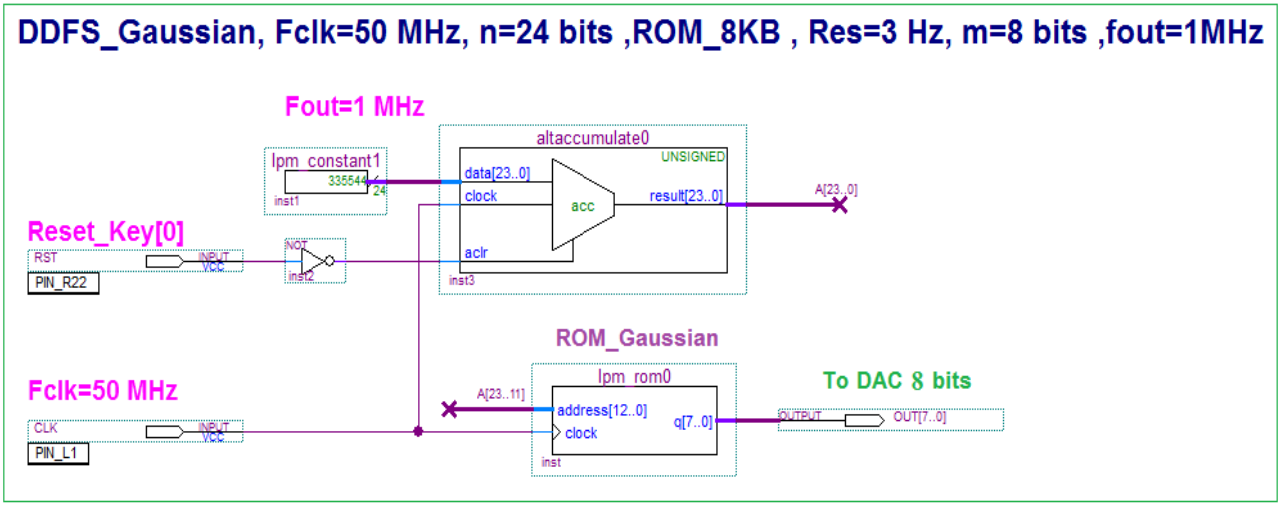

Fig. 9: Block diagram of the Gaussian pulse generator.

The design result of the Gaussian pulse generator is taken from the digital oscilloscope screen and shown on the figure (10-a) for (a =1, f_{out}=1MHz) and on the figure (10-b) for (a =2, f_{out}=1MHz).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

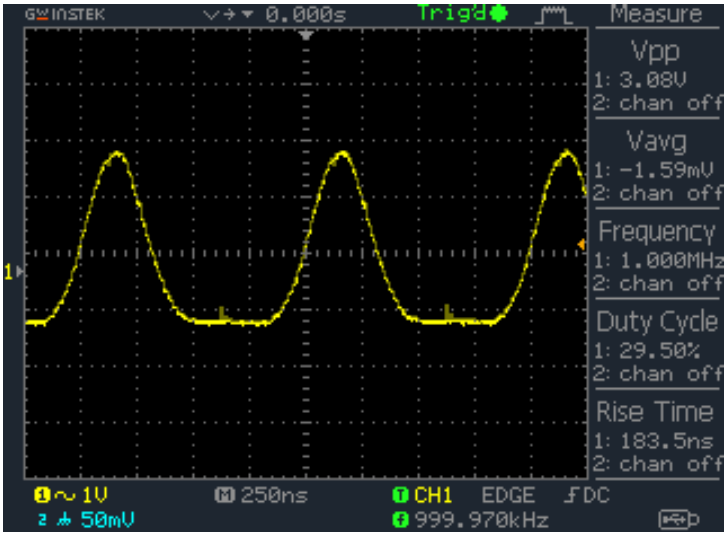

Fig. 10-a: The Gaussian pulse on the digital oscilloscope screen for (a =1, f_{out}=1MHz).

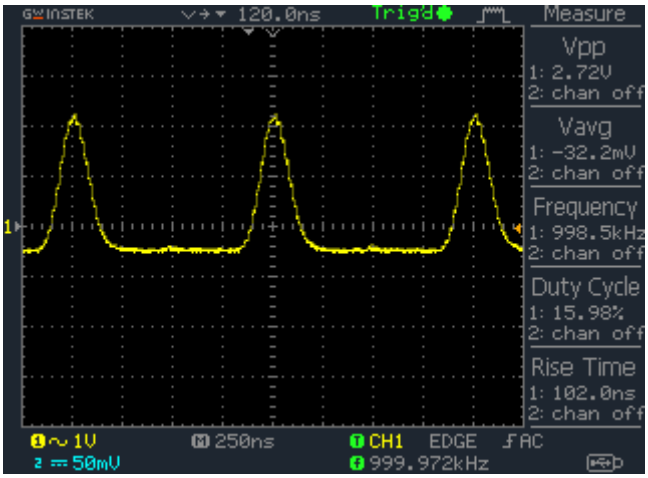

Fig. 10-b: The Gaussian pulse on the digital oscilloscope screen for (a =2, f_{out}=1MHz).

VIII. THE BLOCK DIAGRAM OF THE TRIANGULAR SIGNAL GENERATOR AND SPECIFICATIONS

The values of the triangular signal that are stored in ROM are calculated according to the relationship [5]:
For :

$$\mathrm{m} = 8\,\text{bits}\,,\ \mathrm{b} = 13\,\text{bits}\,,\ i = (0\ldots 2^{b}-1) = (0\ldots 8191)$$

Then:

$$U_{TRI}(i) = \begin{cases} INT(i/16) & ,\ \text{for i} = (0\ldots.4095) \\ 511 - INT(i/16) & ,\ \text{for i} = (4096\ldots.8191) \end{cases} \qquad (6)$$

$$U_{TRI}(i) = (0\ldots..255\ldots..0)$$

To obtain samples of the triangular signal, we use MATLAB according to the following program section:

```
for x=0:1:8191;
if x<=4095
y=floor(x/16);
else x>=4096
```

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

```
y=511-floor(x/16);
end
q(x+1)=y;
r(x+1)=x;
end
q
r
plot(r,q)
grid;
title('A Trianguar signal');
xlabel('Time');
ylabel('Amplitude')
```

Then we store these values in a notepad file and load it into the triangular signal generator's ROM.
The design result of the triangular signal using (MATLAB R2008a) program is shown on the figure (11).

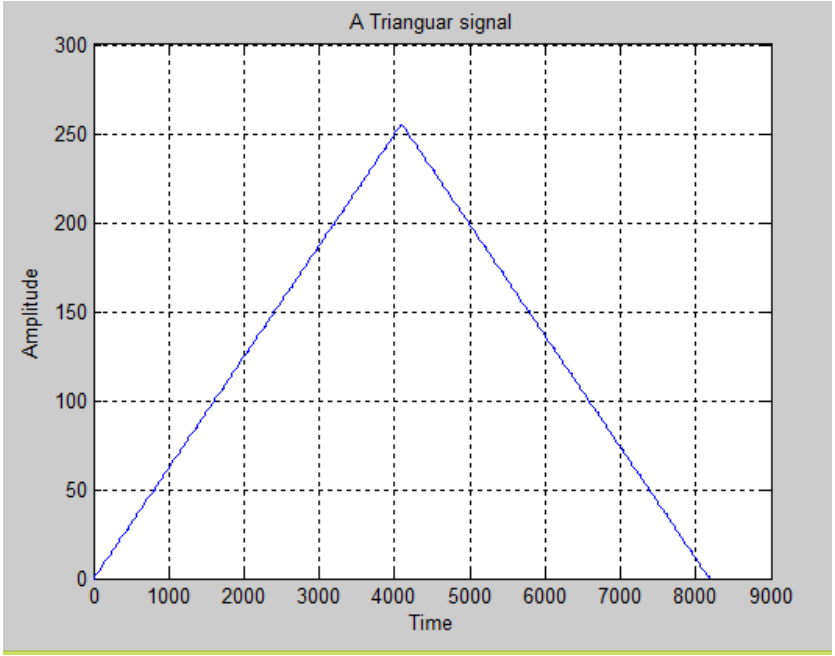

Fig. 11: Triangular signal using MATLAB program.

The block diagram of the Triangular signal Generator in Quartus II 9.1 design environment is shown in figure (12).

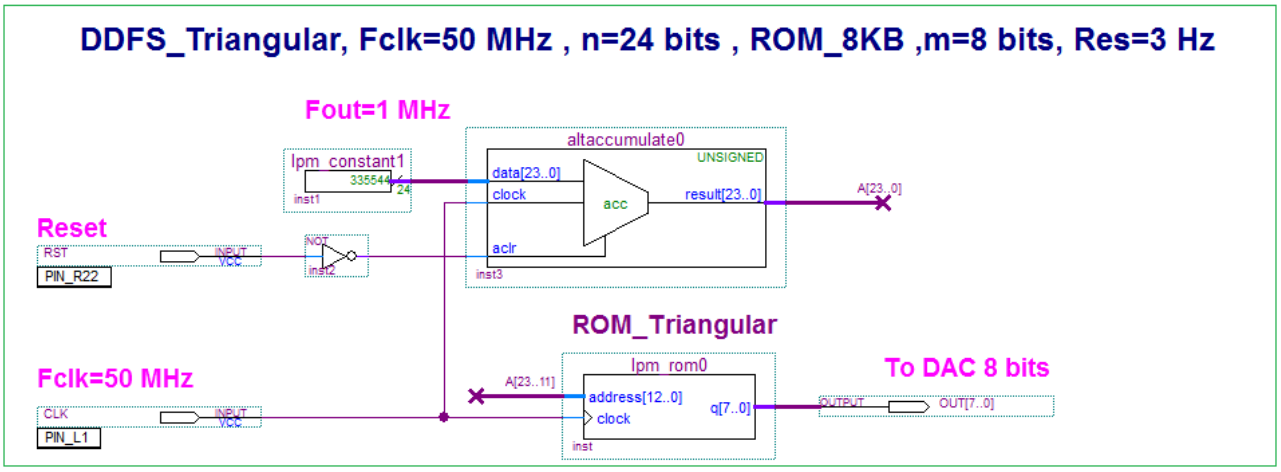

Fig. 12: Block diagram of the Triangular Signal generator.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

The design result of the triangular signal generator is taken from the digital oscilloscope screen and shown on the figure (13) for f_{out}=1MHz.

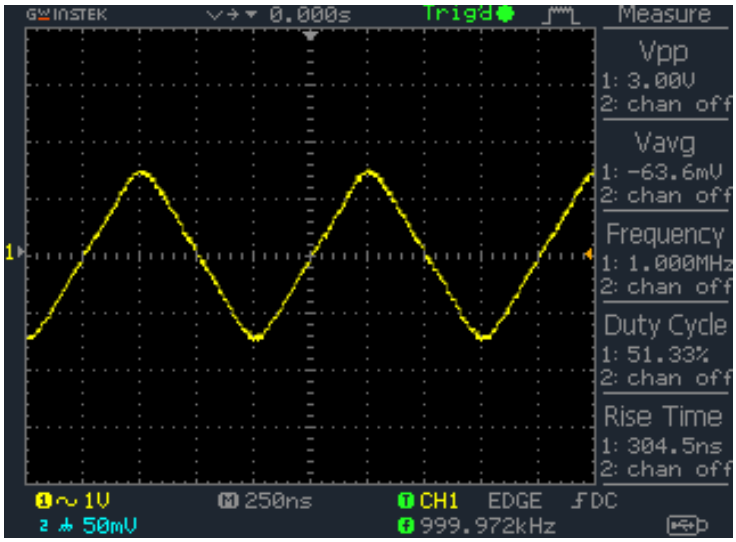

Fig. 13: The triangular signal on the digital oscilloscope screen.

IX. THE BLOCK DIAGRAM OF THE SQUARE PULSE GENERATOR AND SPECIFICATIONS

Depending on the cumulative properties of the phase accumulators in DDS, square pulses can be easily formed without using memories, according to the following functional diagram shown on figure (14).

The period of square pulses (T_{OUT}) is calculated according to the following mathematical relationship:

$$T_{OUT} = \frac{1}{f_{OUT}} = \frac{2^n}{F_{CLK} * L} \Rightarrow L = \frac{2^n}{T_{OUT} * F_{CLK}} \quad (7)$$

For $(f_{OUT} = 1MHz)$:

$$T_{OUT} = \frac{1}{f_{OUT}} = \frac{1}{1} = 1\mu\sec \Rightarrow L = \frac{2^n}{T_{OUT} * F_{CLK}} = L = \frac{2^{24}}{1*50} = 335544$$

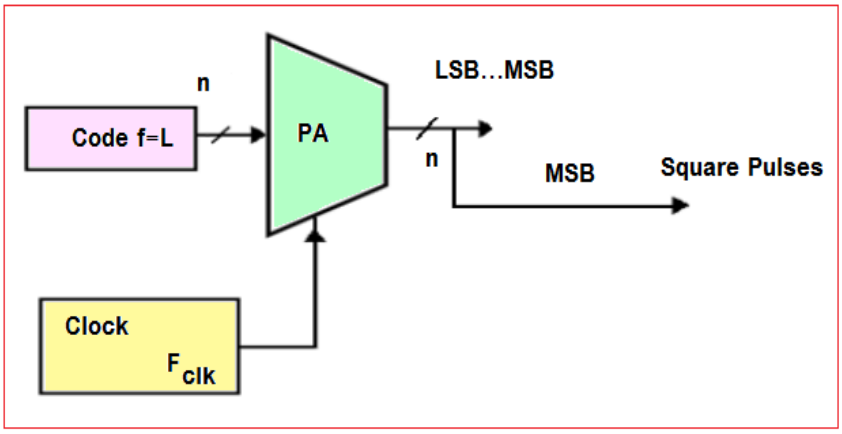

Fig. 14: Functional diagram of digital square generator.

The block diagram of the Square pulse Generator in Quartus II 9.1 design environment is shown in figure (15).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

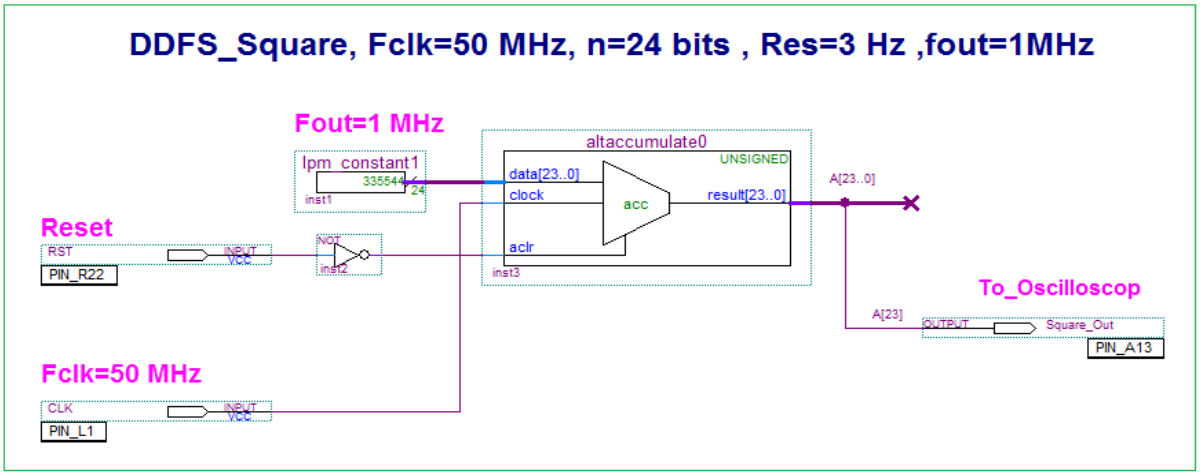

Fig. 15: Block diagram of the Square Pulse Signal generator.

The design result of the square pulse generator is taken from the digital oscilloscope screen and shown on the figure (16) for f_{out}=1MHz.

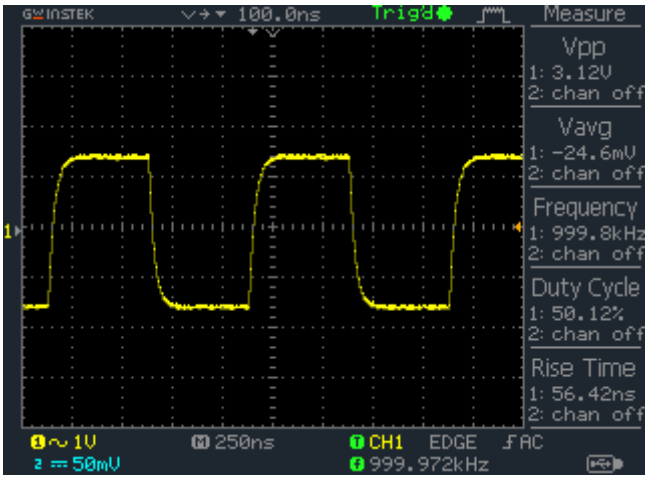

Fig. 16: The square pulse on the digital oscilloscope screen.

X. THE BLOCK DIAGRAM OF THE SAWTOOTH SIGNAL GENERATOR AND SPECIFICATIONS

Depending on the cumulative properties of the phase accumulators in DDS, triangular pulses can be easily formed without using memories, according to the following functional diagram shown on figure (17).

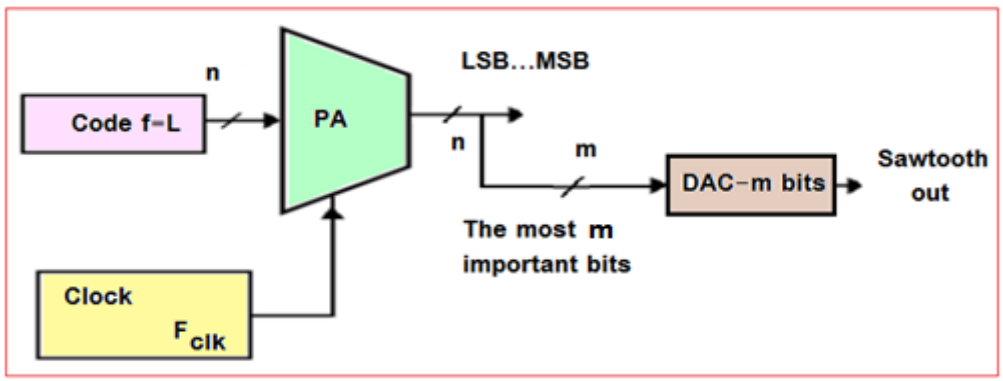

Fig. 17: Functional diagram of digital sawtooth generator.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

The block diagram of the sawtooth signal Generator in Quartus II 9.1 design environment is shown in figure (18).

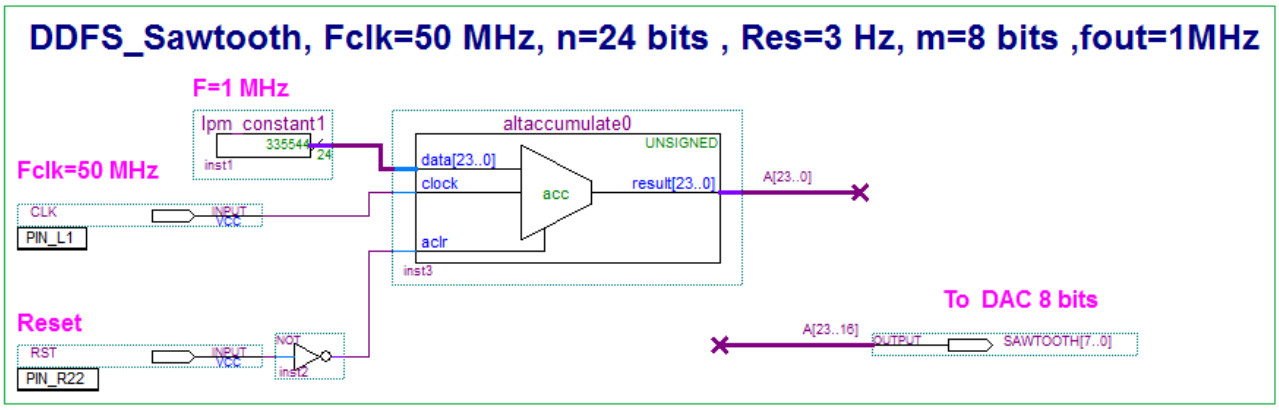

Fig. 18: Block diagram of the Sawtooth Signal generator.

The design result of the sawtooth signal generator is taken from the digital oscilloscope screen and shown on the figure (19) for f_{out}=1MHz.

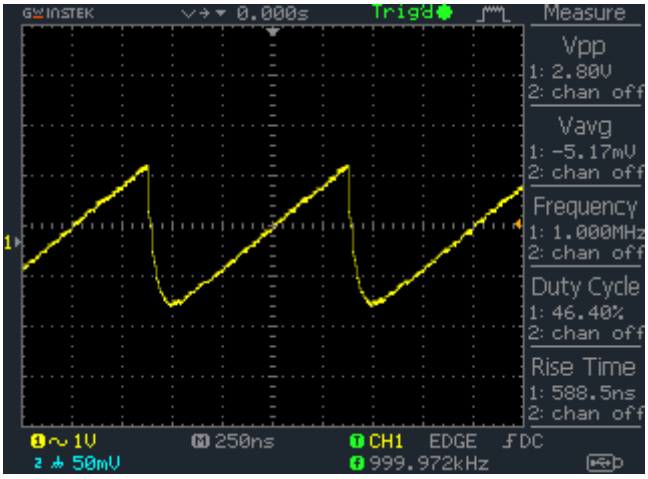

Fig. 19: The sawtooth signal on the digital oscilloscope screen.

XI. THE BLOCK DIAGRAM OF THE DIGITAL PSEUDO-NOISE GENERATOR (DPNG) AND SPECIFICATIONS

To generate the white noise signal, a digital pseudo noise generator DPNG is used, which consists of a shift register of k=60 bits and clock pulses of frequency equals to sampling frequency of 50MHz with maximum periodic time (T_{DPNG}) for the generated series [7]:

$$T_{DPNG} = (2^k - 1) * T_{sam} \qquad (7)$$

$$\text{For k} = 60\,,\ \mathrm{T_{sam}} = \frac{1}{F_{CLK}} = \frac{1}{50} = 0.02\mu s \ \Rightarrow$$

$$T_{DPNG} = (2^{60} - 1) * 0.02 * 10^{-6} = 2.3 * 10^{10}\, s$$

Or:

$$T_{DPNG} = 2.3 * 10^{10} / (3600 * 24 * 365) = 731.2\, Year$$

Where k the number of shift register bits.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

According to the diagram shown in figure. (20), which consists of the following parts:
-Shift register of 60 bits and clock pulses with sampling frequency of 50MHz.
-Feedback circuit by using XOR gate and NOT gate.
The figure. (21) shows the formed noise signal.

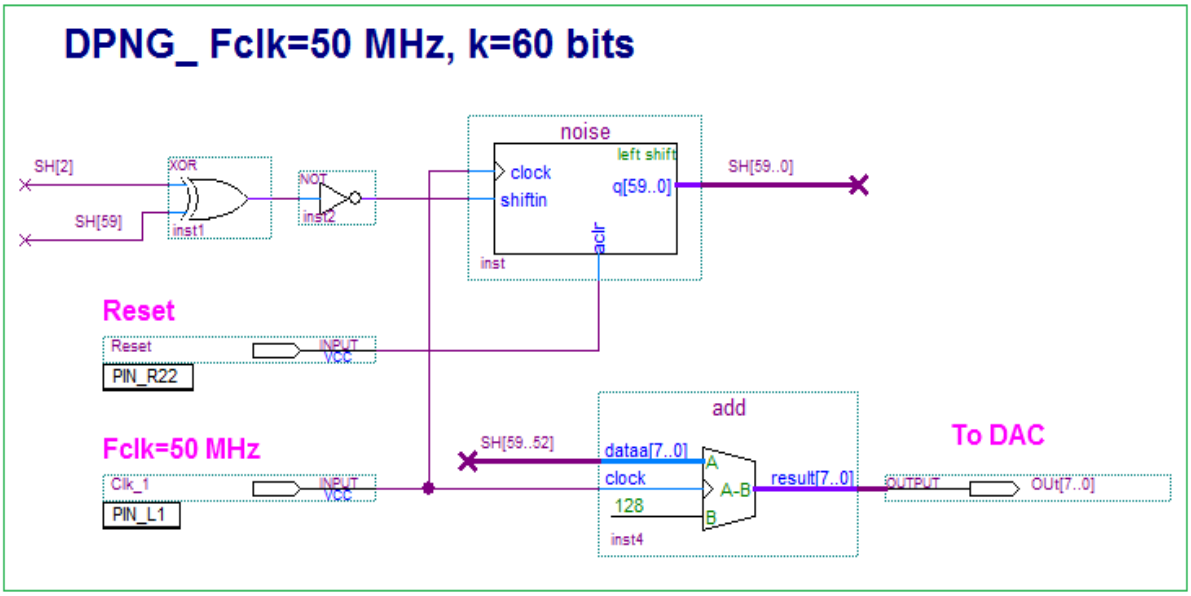

Fig. 20: Block diagram of the Digital Pseudo Noise Generator (DPNG).

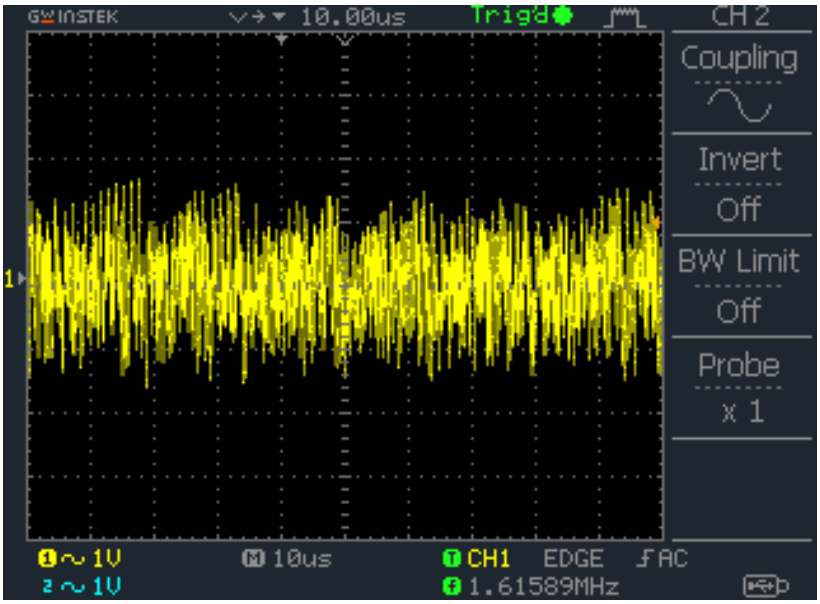

Fig. 21: The white noise signal on the digital oscilloscope screen.

XII. CONCLUSION AND RESULTS

The results of the practical design of a digital signal generator for different types of signals is shown in figure (22).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13503

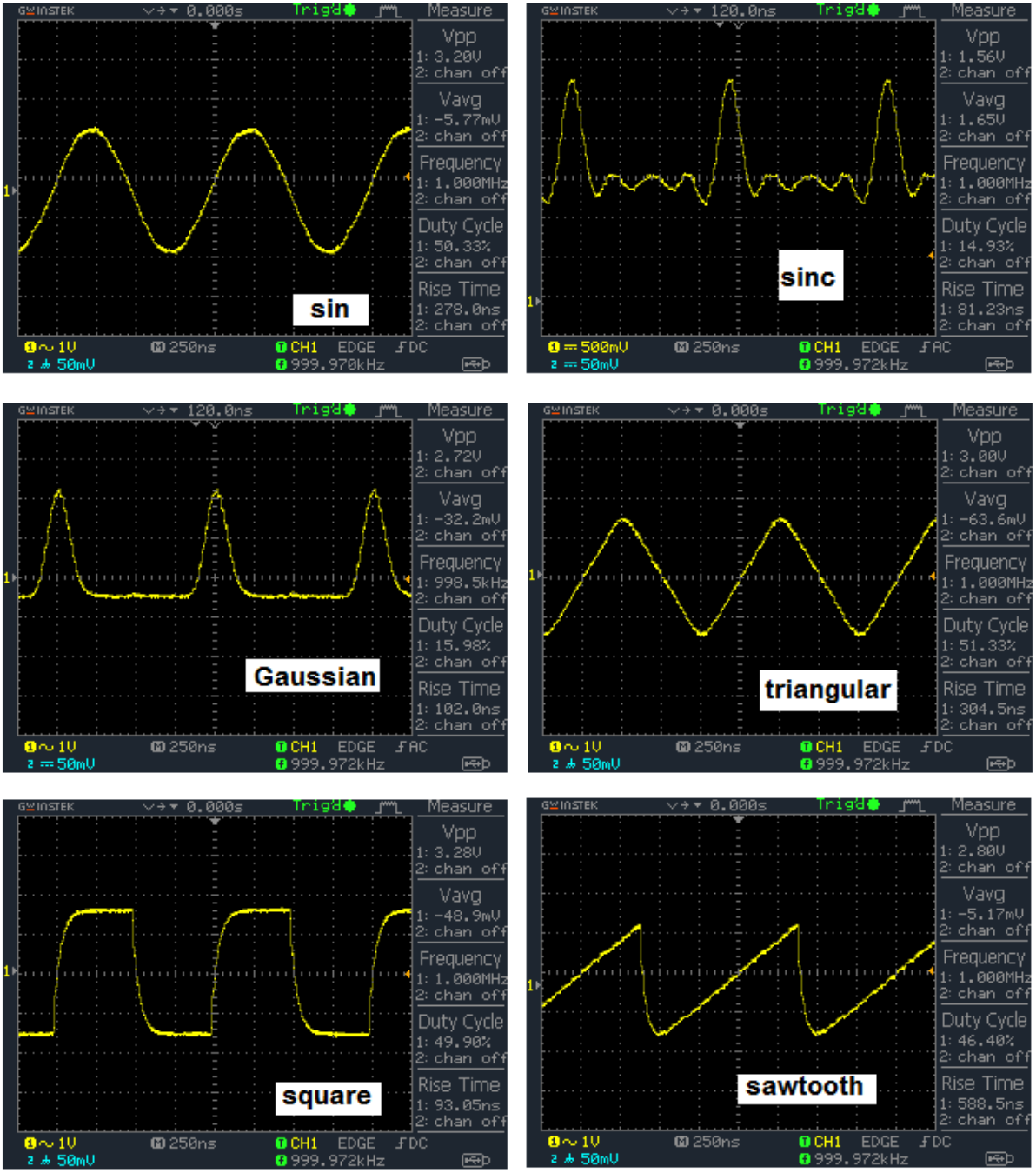

Fig. 22 :The results of the practical design of a DFSG for different types of signals on the digital oscilloscope screen.

Based on the theoretical analysis and practical implementation that took place in this research, some important points can be extracted, which we summarize as follows:

-The use of DDFS allows generating any signal no matter how complex, as well as achieving modulation of any type, and processing of frequency hopping at high speeds.

- The use of FPGA in the design process allows the possibility of modifying the design, amending its specifications and improving them according to demand easily by changing the software design only.

- Any type of signal can be formed with very high accuracy، regardless of its complexity. We only need a formula to describe the signal mathematically, tabular, graphical, or other.

DOI: 10.17148/IJARCCE.2024.13503

REFERENCES

[1] Hemanth Kumar S N , Venkateshappa , Design and Implementation of DDS Module on FPGA, International Journal of Advanced Trends in Computer Science and Engineering , Volume 10, No.2, March - April 2021.
[2] Aswathi Anil, FPGA Implementation of DDS for Arbitrary wave generation ,International Journal of Engineering Research and Applications, Vol. 11, Issue 7, (Series-II) July 2021, pp. 56-64
[3] www.altera.com.
[4] Volnei A. Pedroni, Circuit Design With VHDL, MIT Press Cambridge, Massa- chusetts London, England (2004) 364.
[5] GOLDBERG B. 1999- Digital Frequency Synthesis Demystified, LLH Technology Publishing, united states, 334.
[6] K. Aboutabikh , A. Garib , Design and Implementation of a Digital Matched Filter for Square Pulses Signals using FPGA ,International Journal of Advanced Research in Computer and Communication Engineering Impact Factor 8.102, Peer-reviewed & Refereed journal, Vol. 13, Issue 4, April 2024.
[7] Afaq Ahmad, Sayyid Samir Al-Busaidi and Mufeed Juma Al-Musharafi. On Properties of PN Sequences Generated by LFSR – a Generalized Study and Simulation Modeling. Indian Journal of Science and Technology-2013.
[8] C. S.Rawat, Deepak Balwani , Dipti Bedarkar , Jeetan Lotwani, Harpreet Kaur Saini , Implementation of Barker Code and Linear Frequency Modulation Pulse Compression Techniques in Matlab International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 4, April 2014(105-111).
[9] Zoran Golubi cic, Slobodan Simic c , Aleksa J. Zejak , Design and FPGA implementation of digital pulse compression for band-pass Radar signals, Journal of Electrical Engineering, VOL. 64, NO. 3, 2013, 191–195.
[10] Introduction to matched filters John C. Bancroft CREWES Research Report . Volume 14 (2002).
[11] Steve Winder .2002-Analog and Digital Filter Design ,second edition , Elsevier Science (USA),450.

BIOGRAPHY

Dr. Kamal Aboutabikh holds a PhD in communication engineering in 1988 from the USSR , university of communication in Leningrad , holds a degree assistant professor in 2009 from Aleppo university.

Lecturer at Department of Biomedical Engineering , Al Andalus University For Medical Sciences-Syria ,Tishreen University-Syria ,Corduba Private University- Syria , Kassala University-Sudan and Ittihad Private University- Syria.

Publish a lot of research in the field of digital communication and digital signal processing in the universities of Syria and in the European and Indian journals.

Working in the field of programming FPGA by using VHDL and design of Digital Filters.

Dr. Abdul-Aziz Shokyfeh holds a PhD in telecom engineering in 2017 from Damascus Syria, Faculty of Mechanical and Electrical Engineering Telecommunication and electronic department .

Lecture at Electric engineering, computer engineering, control and automation engineering, telecom and electronic engineering at Damascus University.

Dean of IT college at Ittehad Private University, from 2023 till now. Publish a lot of research in field of antenna in Syrian and European journals. Working in field of PBXs , Network , low current system and solution.

Dr Amer Garib holds a PhD in communication engineering in 1994 from the USSR, university of communication in Leningrad. Lecturer at Department of Faculty of Informatics Engineering , Ittihad Private University- Syria.

Publish a lot of research in the field of digital communication in the universities of Syria and in the Indian journals. Working in the field of Digital Communication Systems and design of Antenna Systems